

\*\*\*\*\*

ATM Forum Document Number: ATM\_Forum/97-0831

\*\*\*\*\*

Title: GFR -- Providing Rate Guarantees with FIFO Buffers to TCP Traffic.

\*\*\*\*\*

Abstract: In this contribution we present analysis and simulation results on controlling TCP rates by buffer allocation. When segment loss is low, TCP throughput depends primarily on the TCP window size, and the round trip time of the connection. As a result, it is possible to control TCP rates with FIFO queuing. We present a buffer management policy called Differential Fair Buffer Allocation that provides loose rate guarantees to SACK TCP sources when the sum of the MCRs is significantly lower than the network capacity. We study the performance of differential fair buffer allocation by simulation.

\*\*\*\*\*

Source:

Rohit Goyal, Raj Jain, Sonia Fahmy, Bobby Vandalore, Shiv Kalyanaraman  
Department of CIS, The Ohio State University (and NASA)

Raj Jain is now at Washington University in Saint Louis, [jain@cse.wustl.edu](mailto:jain@cse.wustl.edu) <http://www.cse.wustl.edu/~jain/>

Sastri Kota  
Lockheed Martin Telecommunications/Astrolink  
1272 Borregas Avenue,  
Bldg B/551 0/GB - 70  
Sunnyvale, CA 94089  
Email: [sastri.kota@lmco.com](mailto:sastri.kota@lmco.com)

Pradeep Samudra,  
Broadband Network Lab  
Samsung Electronics Co. Ltd.  
Samsung Telecom America, Inc.  
1130 E Arapaho, Richardson, TX 75081  
email: [psamudra@telecom.sna.samsung.com](mailto:psamudra@telecom.sna.samsung.com)

\*\*\*\*\*

Date: September 1997, Paris, France

\*\*\*\*\*

Distribution: ATM Forum Technical Working Group Members (AF-TM)

\*\*\*\*\*

Notice: This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.

\*\*\*\*\*

## 1 Introduction

Guaranteed Frame Rate (GFR) is expected to be used between ATM edge devices. For example, IP routers separated by an ATM network can have GFR VCs set up between them for data transfer. Figure 1 illustrates such a case where the ATM cloud connects several LANs and routers. ATM end systems may also establish GFR VCs for connections that can benefit from a minimum throughput guarantee.

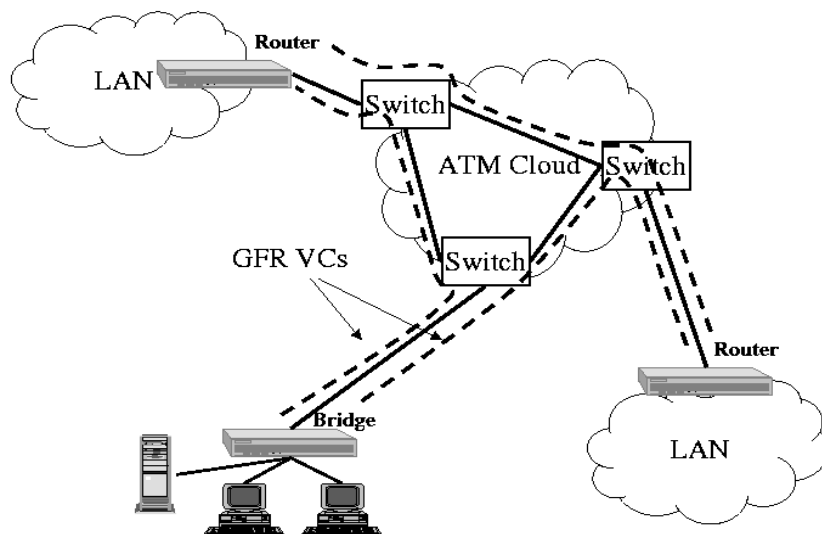


Figure 1: Use of GFR in ATM connected LANs

In the July 1997 meeting, several proposals were made [3, 4, 8] to provide rate guarantees to TCP sources with FIFO queuing in the network. Per-VC scheduling was deemed necessary to provide rate guarantees to TCP connections. However, all these studies were performed at high target network utilization, i.e., most of the network bandwidth was allocated to the GFR VCs. It was mentioned in the July meeting, that rate guarantees should be achievable with a FIFO buffer for low rate allocation. Also, since routers would use GFR VCs, each VC would multiplex many TCP connections through it. All previous studies had examined TCP traffic with a single TCP per VC. Per-VC buffer management for such cases reduces to per-TCP buffer management. For VCs with several aggregated TCPs, per-VC control is unaware of each TCP in the VC. Moreover, aggregate TCP traffic characteristics and control requirements may be different from those of single TCP streams.

In this contribution, we study two main issues:

- Providing minimum rate guarantees to TCP like adaptive traffic with FIFO buffer *for low rate allocations.*
- Traffic management and control of VCs with aggregate TCP flows.

Section 2 discusses the behavior of TCP traffic with controlled windows. This provides insight into controlling TCP rates by controlling TCP windows. Section 3 describes the effect of buffer occupancy and thresholds on TCP throughput. Section 4 presents a simple threshold-based buffer management policy called differential fair buffer allocation to provide TCP throughputs in proportion to buffer thresholds

for low rate allocations. This scheme assumes that each GFR VC may carry multiple TCP connections. We then present preliminary simulation results with differential buffer allocation. *In our simulation and analysis, we use SACK TCP [10] as the TCP model.*

## 2 TCP Behavior with Controlled Windows

TCP uses a window based mechanism for flow control. The amount of data sent by a TCP connection in one round trip is determined by the window size of the TCP connection. The window size is the minimum of the sender's congestion window (CWND) and the receiver's window (RCVWND). As a result, TCP rate can be controlled by controlling the window size of the TCP connection.

Many TCP sources tend to be greedy so that a window limit might not be enforceable by the network to control the TCP rate. TCP sources respond to packet loss by reducing the source congestion window by one-half, and then increasing it by one segment size every round trip. As a result, the average TCP window can be controlled by packet discard at specific CWND values.

Figure 2 shows how the source TCP congestion window varies when a single segment is lost at a particular value of the congestion window. The figure is the CWND plot of the simulation of the configuration shown in Figure 3 with a single TCP source (N=1). The figure shows four different values of the window at which a packet is lost. The round trip latency for the connection is 30 ms.

For window based flow control, the throughput (in Mbps) can be calculated from the average congestion window (in Bytes) and the round trip time (in seconds) as:

$$\text{Throughput (Mbps)} = \frac{8 \times 10^{-6} \times \text{Average CWND (bytes)}}{\text{Round Trip Time (secs)}}$$

The factor  $8 \times 10^{-6}$  converts the throughput from bytes per sec to Megabits per sec. The average TCP CWND during the linear increase phase can be calculated as

$$\text{CWND}_{\text{avg}} = \frac{\sum_{i=1}^T \text{CWND}_{\text{max}}/2 + \text{Max Segment Size} \times i}{T}$$

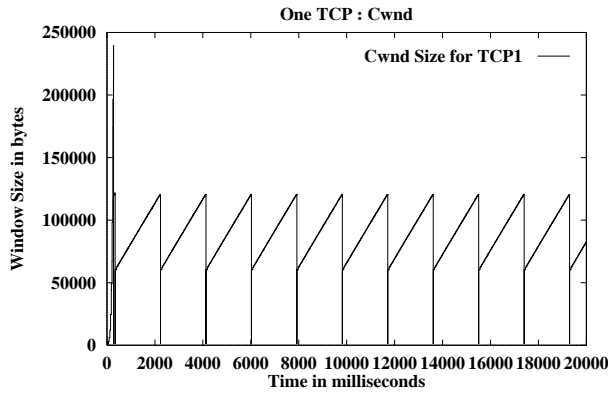
where T is the number of round trip times for the congestion window to increase from  $\text{CWND}_{\text{max}}/2$  to  $\text{CWND}_{\text{max}}$ . Note that this equation assumes that during the linear increase phase, the TCP window increases by one segment every round trip time. However, when the TCP delayed acknowledgment option is set, TCP might only send an ACK for every two segments. In this case, the window would increase by 1 segment every 2 RTTs.

From Figure 2, the average congestion windows in the linear phases of the four experiments are approximately 91232 bytes, 181952 bytes, 363392 bytes and over 600000 bytes. As a result, the average calculated throughputs from the above equation are 24.32 Mbps, 48.5 Mbps, 96.9 Mbps, and 125.6 Mbps (126 Mbps is the maximum possible TCP throughput for a 155.52 Mbps link with 1024 byte TCP segments). The respective throughputs obtained from the simulations of the four cases are 23.64 Mbps, 47.53 Mbps, 93.77 Mbps and 25.5 Mbps. The throughput values calculated from the average congestion windows are close to those obtained by simulation. From this, it appears that controlling the TCP window so as to maintain a desired average window size should enable the network to control the average TCP throughput.

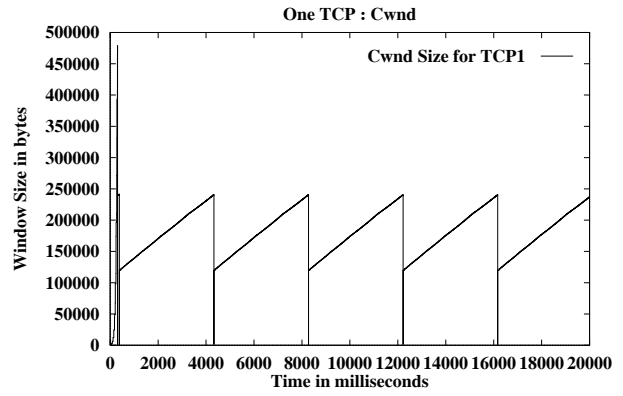
## 3 TCP Window Control using Buffer Management

In the previous section, an artificial simulation was presented where the network controlled the TCP rate by dropping a packet every time the TCP window reached a particular value. In practice, the ATM network

onecpubgfr?option.sack=263/sw\_qlsize.inf=1/sw\_qlsize.inf=1/granularity=100/wnd\_scale\_factor.wan=4/cv\_wm=37500/gfr\_r=0.9/gfr\_z=0.8/sr=50000/stopime.20=1999900/stopime.20=1999900/ms.1024=1024  
ontime=300000/offtime=300000/vbrate=100/time\_int=5000/cr=10/vfrac=0.0/afrac=0.0/v=500/a=1.15/b=1.05/sq=0.5 / Date:08/15/97  
ICR: 10.00 10.00 / XRM: 253.00 253.00 / Graph: 6

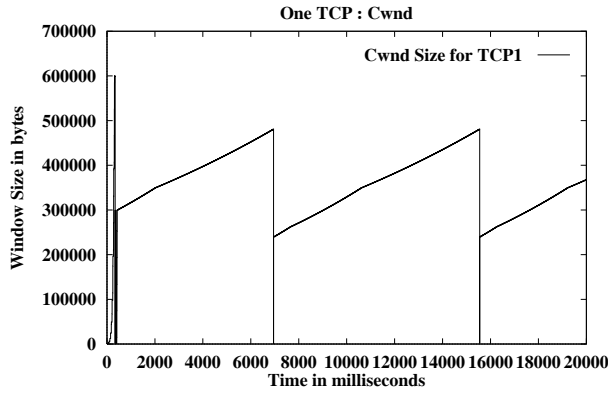


(a)

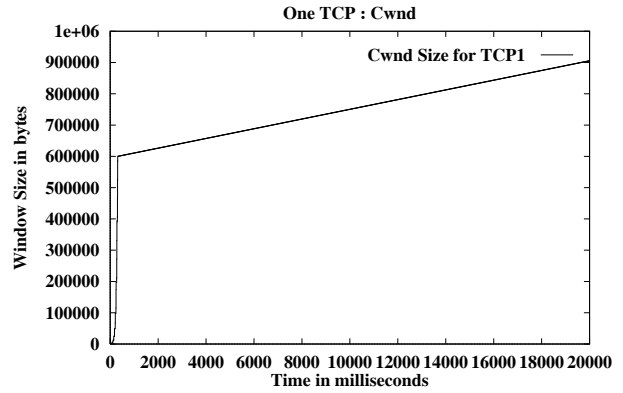


(b)

onecpubgfr?option.sack=263/sw\_qlsize.inf=1/sw\_qlsize.inf=1/granularity=100/wnd\_scale\_factor.wan=4/cv\_wm=37500/gfr\_r=0.9/gfr\_z=0.8/sr=200000/stopime.20=1999900/stopime.20=1999900/ms.1024=1024  
ontime=300000/offtime=300000/vbrate=100/time\_int=5000/cr=10/vfrac=0.0/afrac=0.0/v=500/a=1.15/b=1.05/sq=0.5 / Date:08/15/97  
ICR: 10.00 10.00 / XRM: 253.00 253.00 / Graph: 4



(c)



(d)

Figure 2: Single TCP Congestion Window Control

knows neither the size of the TCP window, nor the round trip time of the connection. A switch can use per-VC accounting of the TCP packets in its buffer to estimate the bandwidth used by the connection.

In a FIFO buffer, the output rate of a connection is determined by the number of packets of the connection in the buffer. Let  $\mu_i$  and  $x_i$  be the output rate and the buffer occupancy respectively of  $VC_i$ . Let  $\mu$  and  $x$  be the total output rate and the buffer occupancy of the FIFO buffer respectively. Then, by the FIFO principle, in steady state,

$$\mu_i = \frac{x_i}{x} \mu$$

or

$$\frac{x_i/x}{\mu_i/\mu} = 1$$

If the buffer occupancy of every active VC is maintained at a desired threshold, then the output rate of each VC can also be controlled. In other words, if a VC always has  $x_i/x$  cells in the buffer, its average output rate will be at least  $\mu x_i/x$ .

Adaptive flows like TCP respond to segment loss by reducing their congestion window. A single packet loss is sufficient to reduce the TCP congestion window by one-half. Consider a drop policy that drops a single TCP packet from a connection every time the connection's buffer occupancy crosses a given threshold. The drop threshold for a connection determines the maximum size to which the congestion window is allowed to grow. Because of TCP's adaptive nature, the buffer occupancy reduces after about 1 RTT. The drop policy drops a single packet when the TCP's buffer occupancy crosses the threshold, and then allows the buffer occupancy to grow by accepting the remainder of the TCP window. On detecting a loss, TCP remains idle for about one-half RTT, during which the buffer occupancy decreases below the threshold. Then the TCP window increases linearly (and so does the buffer occupancy), and a packet is again dropped when the buffer occupancy crosses the threshold. In this way, TCP windows can be controlled quite accurately to within one round trip time. As a result, the TCP's throughput can also be controlled by controlling the TCP's buffer occupancy.

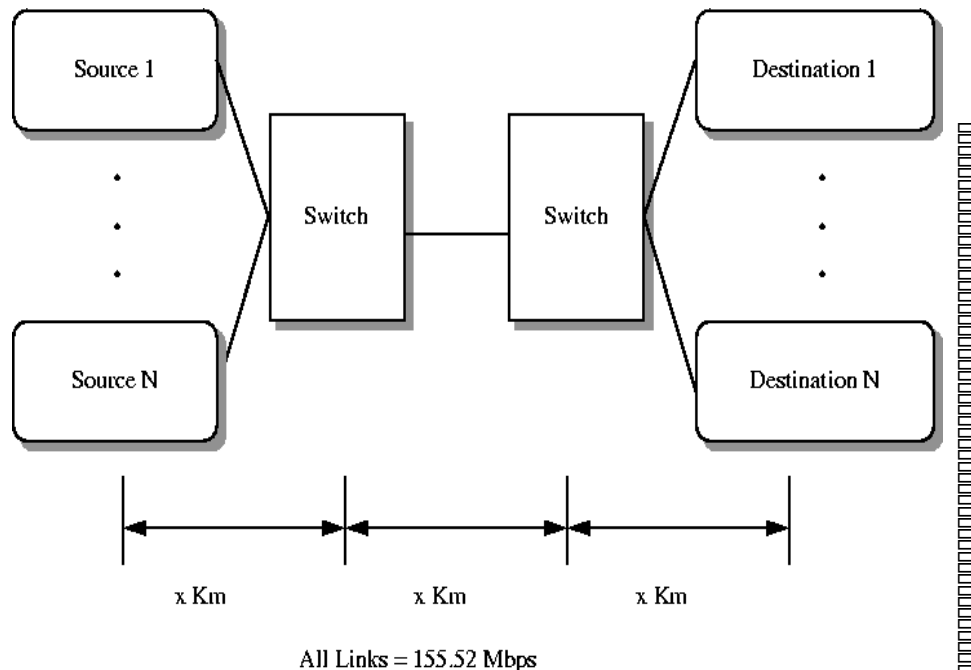


Figure 3:  $N$  source configuration

We performed simulations of the TCP configuration in Figure 3 with fifteen TCP sources. Each TCP

source was a separate UBR VC. Five different buffer thresholds ( $r_i$ ) were selected, and each of three TCP's had the same buffer threshold. Table 1 lists the buffer thresholds for the VC's in the FIFO buffer of the switches. We chose four different sets of thresholds as shown by the threshold columns. The last row in the table shows the total buffer allocated ( $r$ ) to all the TCP connections for each simulation. The total buffer size was large (48000 cells) so that there was enough space for the buffers to increase after the single packet drop. The buffer thresholds were selected in proportion to the SCR of the connection. For each connection, the ratio of the threshold to the total buffer size was proportional to the ratio of the SCR to the PCR. For a 155.52 Mbps link (= 353200 cells per sec approximately), and a buffer size of 48000 cells, the total target utilizations were 29%, 43%, 57%, 71% respectively.

Table 1: Fifteen TCP buffer thresholds

TCP number	Threshold per TCP (cells) ( $r_i$ )			
1-3	305	458	611	764
4-6	611	917	1223	1528
7-9	917	1375	1834	2293
10-12	1223	1834	2446	3057
13-15	1528	2293	3057	3822
Total threshold	13752	20631	27513	34392

Table 2 shows the average throughput obtained per TCP in each threshold category for the four simulations. The TCP throughputs were averaged over each category to reduce the effects of randomness. The last row of the table shows the total throughput obtained in each simulation.

The proportion of the buffer usable by each TCP ( $r_i/r$ ) before the single packet drop should determine the proportion of the throughput achieved by the TCP. Table 3 shows the ratios  $(\mu_i/\mu)/(r_i/r)$  for each simulation. All ratios are close to 1. This indicates that the TCP throughputs are indeed proportional to the buffer allocations. The variations (not shown in the table) from the mean TCP throughputs increased as the total buffer thresholds increased (from left to right across the table). This is because the TCPs suffered a higher packet loss due to the reduced room to grow beyond the threshold. Thus, very high buffer utilization produced more variation in achieved rate (last column of Table 3), whereas in low utilization cases, the resulting throughputs were in proportion to the buffer usage.

Figure 4 shows the congestion windows of one TCP from each TCP category for each of the four simulations. For each simulation, the slopes of the graphs during the linear increase are approximately the same for each TCP, i.e., for a given simulation, the rate of increase of CWND is the same for all TCPs. We know that TCP windows increase by 1 segment every round trip time. Thus, we can conclude that for a given simulation, TCPs sharing the FIFO buffer experience similar queuing delays regardless of the individual per-connection thresholds at which their packets are dropped. This is because, if all TCP's buffer occupancies are close to their respective thresholds, then when a packet arrives at the buffer, it is queued behind cells

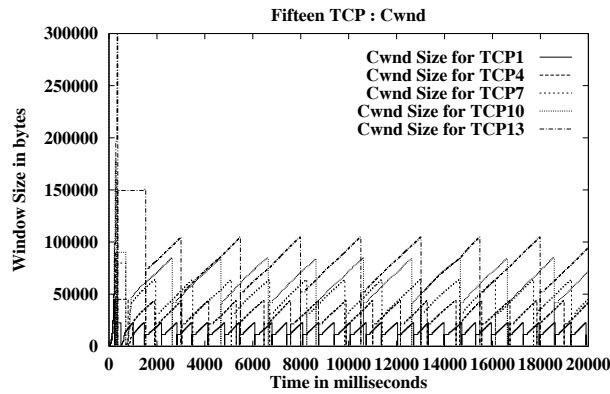
Table 2: Fifteen TCP throughputs

TCP number	Throughput per TCP (Mbps)			
1-3	2.78	2.83	2.95	3.06
4-6	5.45	5.52	5.75	5.74
7-9	8.21	8.22	8.48	8.68
10-12	10.95	10.89	10.98	9.69
13-15	14.34	13.51	13.51	13.93
Total throughput	125.21	122.97	125.04	123.35

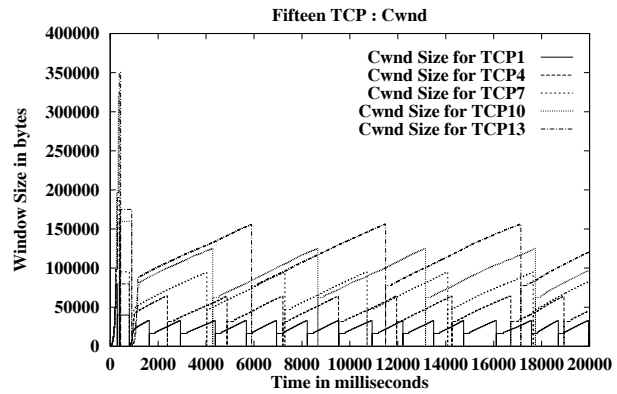
Table 3: Fifteen TCP buffer:throughput ratio

TCP number	$(\mu_i/\mu)/(r_i/r)$			
1-3	1.00	1.03	1.02	1.08
4-6	0.98	1.01	1.03	1.04
7-9	0.98	1.00	1.00	1.02
10-12	0.98	0.99	0.98	0.88
13-15	1.02	0.98	0.97	1.01

15tcpbfgrwan/optmon.gfndrop=2097921,optmon sack=2633w\_gsize inf=1lw\_gsize inf=1lw\_gsize wan=48000/granularity=100/wd\_scale\_factor wan=4rcv\_w in wan=37500/gf\_r=0.9/gf\_r=1.5/gf\_w=0.75/kr=10/optmon.20=19999000  
 sroptime.20=19999000ms.1024=1024ofltime=300000/brate=100time\_jar=5000/0sw\_jar=500/frac=0.0/frac=0.0/0=500=-1.15h=1.05/qh=0.5 / Date:08/26/97  
 sroptime.20=19999000ms.1024=1024ofltime=300000/brate=100time\_jar=5000/0sw\_jar=500/frac=0.0/frac=0.0/0=500=-1.15h=1.05/qh=0.5 / Date:08/26/97

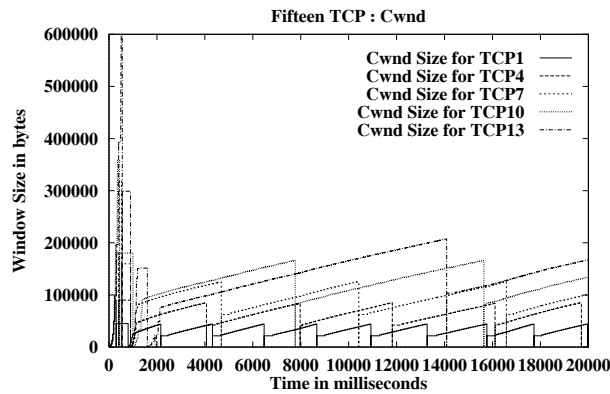


(a)

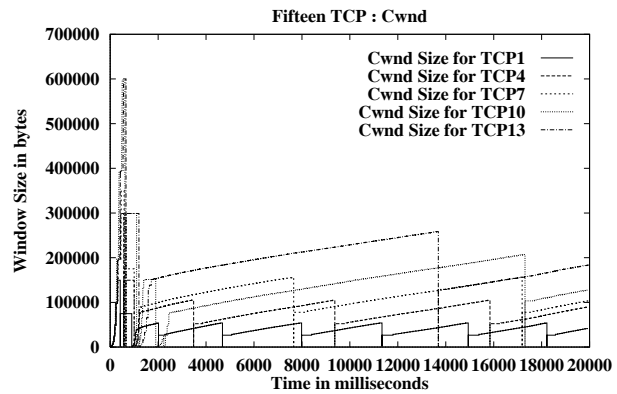


(b)

15tcpbfgrwan/optmon.gfndrop=2097921,optmon sack=2633w\_gsize inf=1lw\_gsize inf=1lw\_gsize wan=48000/granularity=100/wd\_scale\_factor wan=4rcv\_w in wan=37500/gf\_r=0.9/gf\_r=1.5/gf\_w=1.25/kr=10/optmon.20=19999000  
 sroptime.20=19999000ms.1024=1024ofltime=300000/brate=100time\_jar=5000/0sw\_jar=500/frac=0.0/frac=0.0/0=500=-1.15h=1.05/qh=0.5 / Date:08/26/97  
 sroptime.20=19999000ms.1024=1024ofltime=300000/brate=100time\_jar=5000/0sw\_jar=500/frac=0.0/frac=0.0/0=500=-1.15h=1.05/qh=0.5 / Date:08/26/97



(c)



(d)

Figure 4: 15 TCP rate control by packet drop

from  $\Sigma(\text{threshold})$  packets, regardless of the connection to which it belongs. Consequently, each TCP experiences the same average queuing delay.

However, as the total buffer threshold increases, the round trip time for each TCP increases because of the larger total queue size. The larger threshold also results in a larger congestion window at which a packet is dropped. A larger congestion window means that TCP can send more segments in one round trip time. But since the round trip time also increases proportionally to the increase in CWND (because all 15 TCPs are bottlenecked at the first switch), the average throughput achieved by a single TCP remains almost the same (see table 2) across the simulations.

The following list summarizes the discussion so far:

1. TCP throughput can be controlled by controlling its congestion window, which, in turn, can be controlled by setting buffer thresholds to drop packets. This statement clearly assumes that in cases where the offered load is low, and a queue is never built up, then the TCP is allowed to use as much capacity as it can.
2. With a FIFO buffer, the average throughput achieved by a connection is proportional to the fraction of the buffer occupancy that is consumed by the connection's cells.
3. As long as the fraction of buffer occupancy of a TCP can be controlled, its relative throughput is independent of the total number of packets in the buffer, and depends primarily on the fraction of packets of that TCP in the buffer.
4. At a very high buffer utilization, packets may be dropped due to buffer unavailability. This results in larger variations in TCP throughputs. At very high thresholds, the queuing delay also increases significantly, and may cause the TCP sources to timeout.
5. At very low buffer thresholds (high loss rates), TCP sources become unstable and tend to timeout. Also, very low buffer occupancies result in low network utilization. Since TCP can maintain a flow of 1 CWND worth of packets each round trip time, a total buffer occupancy of 1 bandwidth-delay product should provide good utilization.

## 4 Differential Fair Buffer Allocation

The differential fair buffer allocation scheme uses the ideas from the previous sections to provide soft rate guarantees to SACK-TCP like adaptive traffic on ATM connections under low network load conditions. Simulation results of heterogeneous TCP and non-TCP environments will be presented in a future contribution. The policy assumes that multiple TCP connections are multiplexed on a single VC. In this section we present the preliminary design and simulation results of differential fair buffer allocation. A parameter study and sensitivity analysis will be presented in a future contribution. *We assume a model in which TCPs may be merged into a single VC, in which case, the cells of different frames within a VC are not interleaved.* This allows the network to drop frames without having to identify the source that generated the frame.

The switch output port consists of a FIFO buffer for the UBR class with the following attributes:

- **K:** Buffer size in cells.
- **R:** Congestion threshold in cells ( $0 \leq R \leq K$ ). EPD is performed when buffer occupancy is greater than R.



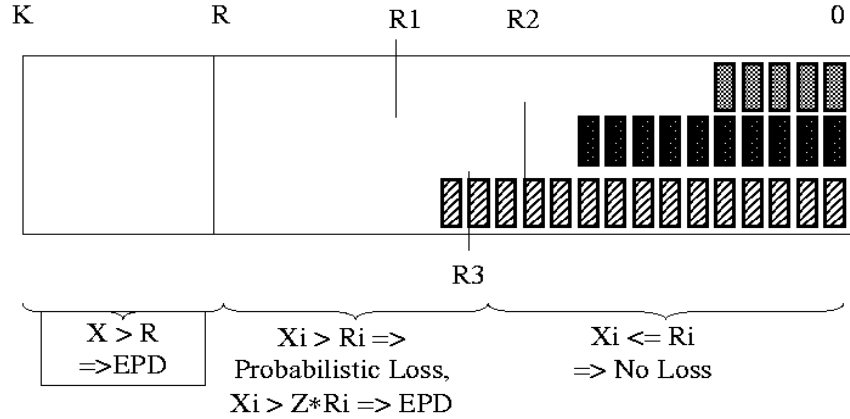


Figure 5: Drop behavior of differential fair buffer allocation

- **Wi:** Weight of VC $_i$  (for example  $W_i = \text{MCR}_i / (\text{Total UBR capacity})$ ).  $\sum W_i < 1$
- **Ri:** Threshold for VC $_i$ . Here we use  $R_i = W_i * R$
- **X:** Number of cells in the buffer.
- **Xi:** Number of cells of VC $_i$  in the buffer.
- **Z:** Parameter ( $Z > 1$ ). We selected  $Z = 1.5$  in our simulations.
- **u:** Uniform(0,1) random number.

When the first cell of a frame arrives at the buffer, if the number of cells ( $X_i$ ) of VC $_i$  in the buffer is less than its threshold ( $R_i$ ), then the cell and frame is accepted into the buffer. If  $X_i$  is greater than  $R_i$ , and if the total buffer occupancy ( $X$ ) is greater than the buffer threshold ( $R$ ), or if  $X_i$  is greater than  $Z \times R_i$ , then the cell and frame are dropped (EPD). Thus  $Z$  specifies a maximum per-VC buffer occupancy during congestion periods. Under low or mild load conditions,  $R \times Z$  should be large enough to buffer a burst of cells without having to perform EPD. If the  $X_i$  is greater than  $R_i$ , and  $X$  is less than  $R$ , then the cell/frame are dropped in a probabilistic manner. The probability of frame drop depends on how much  $X_i$  is above  $R_i$ , as well as the weight ( $W_i$ ) of the connection. As  $X_i$  increases beyond  $R_i$ , the probability of drop increases. Also, the drop probability is higher for connections with a higher threshold. This is because, TCP flows with higher windows (due to higher thresholds) are more robust to packet loss than TCP flows with lower windows. Moreover, in the case of merged TCPs over a single VC, VCs with a high threshold are likely to carry more active TCP flows than those with a low threshold. As a result, a higher drop probability is more likely to hit more TCP sources and improve the fairness within a VC.

The frame is dropped with a probability

$$P\{\text{drop}\} = W_i \times \frac{X_i - R_i}{Z \times R_i - R_i}$$

The drop probability may be further scaled depending on the desired level of control. In addition, if  $X_i$  is greater than  $R_i$ , then all tagged frames may also be dropped. Tagging support is not yet tested for this drop policy.

The resulting algorithm works as follows. When the first cell of a frame arrives:

```

IF ( $X_i < R_i$ ) THEN

    ACCEPT CELL AND FRAME

ELSE IF (( $X < R$ ) AND
        (Cell NOT Tagged) AND
        ( $u > W_i \cdot (X_i - R_i) / (R_i(Z-1))$ ))

    THEN

        ACCEPT CELL AND FRAME

ELSE

    DROP CELL AND FRAME

ENDIF
    
```

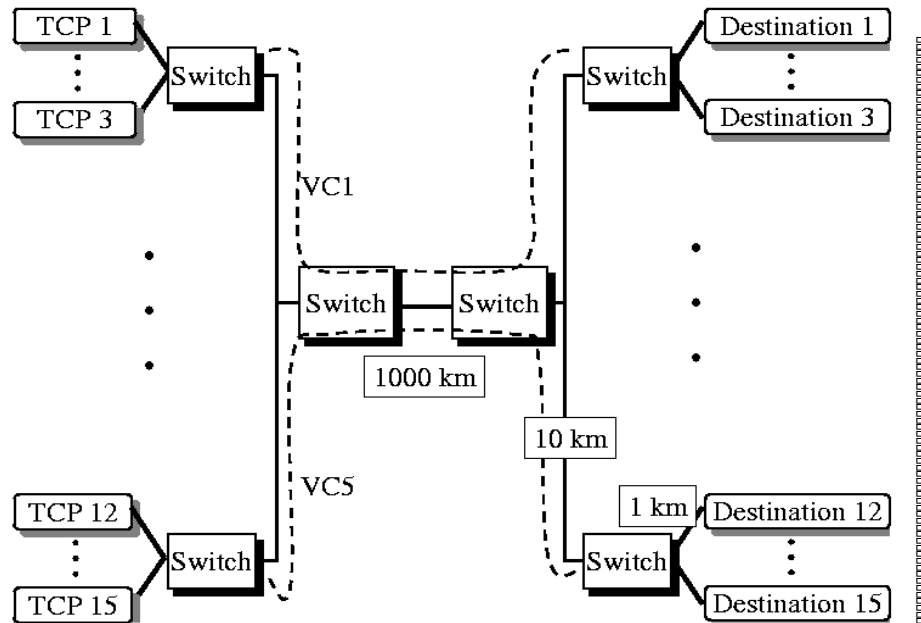


Figure 6: N source VC merge configuration

Figure 6 illustrates the 15 TCP configuration in which groups of three TCPs are merged into 1 single VC. Each local switch merges the 3 TCPs into a single VC. The backbone link has 5 VCs going through it, each with 3 TCPs. The local switches ensure that the cells of frames within a single VC are not interleaved. All the switches implement the differential fair buffer allocation policy described above. The local switches use a separate threshold for each TCP, while the backbone switches use a different threshold for each VC.

Table 4: Differential FBA thresholds

VC number	Threshold (cells)		
1	152	305	611
2	305	611	1223
3	458	917	1834
4	611	1223	2446
5	764	1528	3057
Total	2290	4584	9171

Table 5: Differential FBA simulation results

VC number	Ratio $(\mu_i/\mu)/(r_i/r)$		
1	1.04	1.01	1.16
2	1.05	1.02	1.06
3	0.97	0.99	1.05
4	0.93	1.00	1.13
5	1.03	0.99	0.80

We simulated the 15 merged TCP configuration with 3 different buffer threshold sets. The parameter  $Z$  was set to 1.5, therefore, EPD was performed for each VC when its buffer occupancy was  $1.5 \times R$ . Table 4 shows the thresholds used for each VC at the first bottleneck switch.

Table 5 shows the ratio  $(\mu_i/\mu)/(r_i/\Sigma r_i)$  for each VC for the configuration in Figure 6 and the corresponding thresholds. In all cases, the achieved link utilization was almost 100%. The table shows that TCP throughputs obtained were in proportion to the buffers allocated (since most of the ratios in table 5 are close to 1). The highest variation (not shown in the table) was seen in the last column because of the high threshold values.

In our simulations, the maximum observed queue sizes in cells in the first backbone switch (the main bottleneck) were 3185, 5980 and 11992 respectively. The total allocated buffer thresholds were 2230, 4584 and 9171 cells for the experiments. For a given buffer size, the allocated thresholds represent the SCRs of the connections. At higher rate allocations, the drop policy cannot provide tight bounds on throughput.

Ideally, differential buffer allocation should allocate all excess capacity in proportion to the buffer allocations. The results in Table 5 seem to suggest so, especially for low allocations. In all cases, the total TCP throughput is over 95% of the total link capacity. However, excess bandwidth may not always be allocated linearly in proportion to the buffer allocations. The fairness properties of differential fair buffer allocation are a topic of further study.

## 5 Summary and Future Work

In this contribution, we have used FIFO buffers to attempt to control SACK TCP rates by differential buffer allocation. An optimal set of thresholds should be selected that is high enough to provide sufficient network utilization, and is low enough to allow stable operation. The achieved TCP throughputs are in proportion to the fraction of the average buffer occupied by the VC.

Much work remains to be done to further modify differential fair buffer allocation to work with a variety of configurations. In particular:

- We have only studied the performance of differential fair buffer allocation with SACK TCP. Its

performance with heterogeneous TCPs is a topic of further study.

- We have not studied the effect of non adaptive traffic (like UDP) on the drop policy. It appears that for non adaptive traffic, the thresholds must be set lower than those for adaptive traffic (for the same MCR), and the dropping should be more strict when the buffer occupancy crosses the threshold.
- More simulation configurations need to be studied, including TCP with different RTTs and topologies.
- In this contribution we have not studied the effect of network based tagging in the context of GFR. In the strict sense, GFR only provides a low CLR guarantee to the CLP=0 cell stream i.e., the cells that were not tagged by the source and passed the GCRA conformance test. However, when source (this could be a non-ATM network element like a router) based tagging is not performed, it is not clear if the CLP0 stream has any significance over the CLP1 stream. Moreover, network tagging is an option that must be signaled during connection establishment.

## References

- [1] ATM Forum, "ATM Traffic Management Specification Version 4.0," April 1996, <ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.ps>
- [2] Debashis Basak, Surya Pappu, "TCP over GFR Implementation with Different Service Disciplines: A Simulation Study"
- [3] Debashis Basak, Surya Pappu, "GFR Implementation Alternatives with Fair Buffer Allocation Schemes," ATM Forum 97-0528.
- [4] Olivier Bonaventure, "A simulation study of TCP with the proposed GFR service category," DAGSTUHL Seminar 9725, High Performance Networks for Multimedia Applications, June 1997, Germany
- [5] John B. Kenney, "Satisfying UBR+ Requirements via a New VBR Conformance Definition," ATM FORUM 97-0185.
- [6] John B. Kenney, "Open Issues on GFR Frame Discard and Frame Tagging," ATM FORUM 97-0385.
- [7] Juha Heinanen, and Kalevi Kilkki, "A fair buffer allocation scheme," Unpublished Manuscript.
- [8] R. Goyal, R. Jain et.al, "Simulation Experiments with Guaranteed Frame Rate for TCP/IP Traffic," ATM Forum 97-0607.
- [9] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy and Seong-Cheol Kim, "UBR+: Improving Performance of TCP over ATM-UBR Service," Proc. ICC'97, June 1997.
- [10] R. Goyal, R. Jain et.al., "Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks," To appear, Proc. IC3N'97, September 1997. <sup>1</sup>
- [11] Roch Guerin, and Juha Heinanen, "UBR+ Service Category Definition," ATM FORUM 96-1598, December 1996.
- [12] Roch Guerin, and Juha Heinanen, "UBR+ Enhancements," ATM FORUM 97-0015, February 1997.

---

<sup>1</sup>All our papers and ATM Forum contributions are available from <http://www.cis.ohio-state.edu/~jain>